Mechanics and Mechanical Engineering
Vol. 18, No. 2 (2014) 135–149

Identification of the Mathematical Model of an Underwater Robot Using Artificial Inteligence

Józef Giergiel
Krzysztof Kurc
Dariusz Szybicki
Department of Applied Mechanics and Robotics
Rzeszow University of Technology
bartek@prz.edu.pl
kkurc@prz.edu.pl
dszybicki@prz.edu.pl

Received (10 September 2014)
Revised (16 September 2014)
Accepted (25 September 2014)

The paper focuses on the comparison of identification of the mathematical model of an underwater robot by making use of fuzzy logic systems and neural networks. The solution to the problem was carried out through simulations.

Keywords: Modeling, identification, neural network, fuzzy logic, robot.

1. Introduction

It is hard to take all phenomena into consideration when modelling manipulators or robots, therefore the corresponding mathematical models are not known exactly. Correct analysis of dynamics of such complex systems requires identification of dynamical equations of motion [4, 8]. The identification of mathematical models with the use of neural networks and fuzzy logic systems [1, 5] enables one to recognize unknown parameters and adjust the mathematical model to the real object.

2. Identification using fuzzy logic

In the design of fuzzy sets, a the most important is specification of the consideration set. In the case of an ambiguous term "high temperature", another value will be considered too high, if we accept the temperature interval [0÷10^{o}C], and other, if we accept the temperature interval [0÷1000^{o}C]. The consideration domain or the set of action will be marked with the X letter. We must remember that X is a fuzzy set.

The definition of the fuzzy set [12] was formulated as follows:

A fuzzy set in non–empty space X, written down as $A \subseteq X$, is called the set of pairs

$$A = \{(x, \mu_A(x)); x \in X\} \tag{1}$$

where:

$$\mu_A : X \to [0, 1] \tag{2}$$

is a membership function of the A fuzzy set. This function assigns for every element $x \in X$ some affiliation degree to the fuzzy set A. Three cases are distinuished: full affiliation of the element x to the fuzzy set A, when $\mu_A(x) = 1$, no affiliation of the element x to the fuzzy set A, when $\mu_A(x) = 0$, and partial affiliation to the fuzzy set A of the element x, when $0 < \mu_A(x) < 1$. There are many standard forms of the membership function, which have been described in literature, e.g. [12], however the most common are: gauss functions, triangular functions and trapezoidal functions [3, 11].

In systems with fuzzy logic the rules are symbolic "**IF–THEN**", quality variables are described with linguistic variables and there are fuzzy operators like "**AND**", so the sample rule can be written as follows

$$\textbf{IF } x1 \text{ is small } \textbf{AND } x2 \text{ is large } \textbf{THEN } y \text{ is average} \tag{3}$$

A mathematical model [6, 7, 9, 10] was adopted (Fig. 1b, 1c) for the description of the movement of the underwater robot (Fig. 1a).

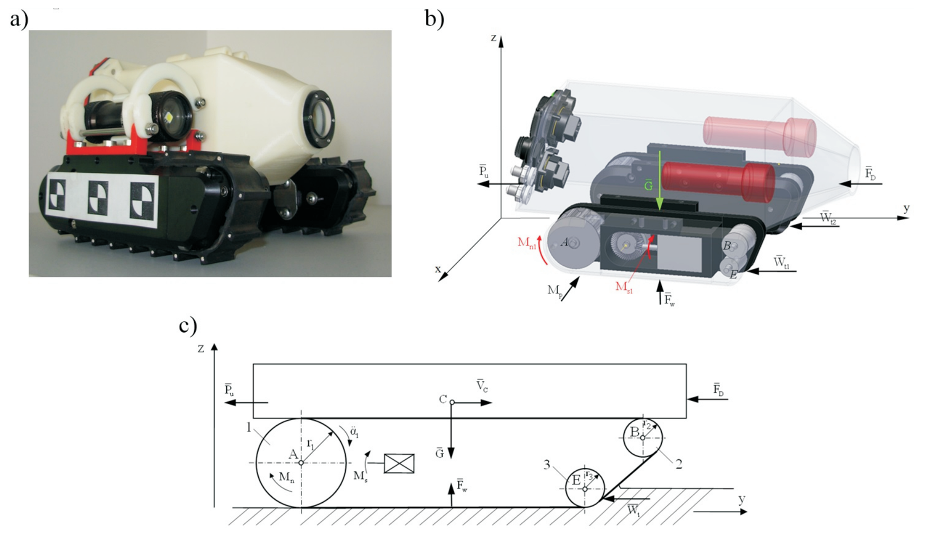

Figure 1 a) Underwater robot, b), c) model of the robot

The dynamic equations of motion is

$$\begin{cases} M_{n1} = & a_4\ddot{\alpha}_1 + a_2\dot{\alpha}_1^2 + a_3\ddot{\alpha}_2 + a_1\dot{\alpha}_2^2 - a_5 \\ M_{n2} = & b_4\ddot{\alpha}_1 + b_2\dot{\alpha}_1^2 + b_3\ddot{\alpha}_2 + b_1\dot{\alpha}_2^2 - b_5 \end{cases} \tag{4}$$

where:

$$
\begin{aligned}
a_1 &= \frac{r^3\left(1-s_2\right)^2\cos\left(\beta\right)\left(m_R+2m\right)\left(1-s_1\right)\sin\left(\beta\right)}{4H} \\
&\quad -\frac{r^3\left(1-s_2\right)^2\sin\left(\beta\right)\cos\left(\gamma\right)^2\left(m_R+2m\right)\left(1-s_1\right)\cos\left(\beta\right)}{4H} \\
a_2 &= \frac{r^3\left(1-s_1\right)^3\cos\left(\beta\right)\left(m_R+2m\right)\sin\left(\beta\right)}{4H} \\
&\quad +\frac{r^3\left(1-s_2\right)^3\sin\left(\beta\right)\cos\left(\gamma\right)^2\left(m_R+2m\right)\cos\left(\beta\right)}{4H} \\
a_3 &= \frac{1}{4}r^2\left(1-s_2\right)\sin\left(\beta\right)^2\left(m_R+2m\right)\left(1-s_1\right) \\
&\quad +\frac{1}{4}r^2\left(1-s_2\right)\cos\left(\beta\right)^2\cos\left(\gamma\right)^2\left(m_R+2m\right)\left(1-s_1\right) \\
&\quad \frac{1}{4}r^2\left(1-s_2\right)\sin\left(\gamma\right)^2\left(m_R+2m\right)\left(1-s_1\right) \\
&\quad -\frac{r^2\left(1-s_2\right)\left(I_R+2I_z+2mH^2\right)\left(1-s_1\right)}{H^2} \\
a_4 &= \frac{1}{4}r^2\left(1-s_1\right)^2\sin\left(\beta\right)^2\left(m_R+2m\right) \\
&\quad +\frac{1}{4}r^2\left(1-s_1\right)^2\cos\left(\beta\right)^2\cos\left(\gamma\right)^2\left(m_R+2m\right) \\
&\quad \frac{1}{4}r^2\left(1-s_1\right)^2\sin\left(\gamma\right)^2\left(m_R+2m\right)+\frac{r^2\left(1-s_1\right)^2\left(I_R+2I_z+2mH^2\right)}{H^2}+I_x \\
a_5 &= r\left(1-s_1\right)\left[0,5F_w\sin\left(\gamma\right)-W_{t1}-0,5P_u-0,5F_D-0,5G\sin\left(\gamma\right)\right] \\
&\quad +\frac{M_p r\left(1-s_1\right)}{H} \\
b_1 &= \frac{r^3\left(1-s_2\right)^3\cos\left(\beta\right)\left(m_R+2m\right)\sin\left(\beta\right)}{4H} \\
&\quad -\frac{r^3\left(1-s_2\right)^3\sin\left(\beta\right)\cos\left(\gamma\right)^2\left(m_R+2m\right)\cos\left(\beta\right)}{4H} \\
b_2 &= \frac{r^3\left(1-s_1\right)^2\cos\left(\beta\right)\left(m_R+2m\right)\left(1-s_2\right)\sin\left(\beta\right)}{4H} \\
&\quad +\frac{r^3\left(1-s_1\right)^2\sin\left(\beta\right)\cos\left(\gamma\right)^2\left(m_R+2m\right)\left(1-s_2\right)\cos\left(\beta\right)}{4H} \\
b_3 &= \frac{1}{4}r^2\left(1-s_2\right)^2\sin\left(\beta\right)^2\left(m_R+2m\right) \\
&\quad +\frac{1}{4}r^2\left(1-s_2\right)^2\cos\left(\beta\right)^2\cos\left(\gamma\right)^2\left(m_R+2m\right) \\
&\quad \frac{1}{4}r^2\left(1-s_2\right)^2\sin\left(\gamma\right)^2\left(m_R+2m\right)+\frac{r^2\left(1-s_2\right)^2\left(I_R+2I_z+2mH^2\right)}{H^2}+I_x
\end{aligned}
$$

$$\begin{aligned}
b_4 &= \frac{1}{4}r^2(1-s_2)\sin(\beta)^2(m_R+2m)(1-s_1) \\
&\quad +\frac{1}{4}r^2(1-s_2)\cos(\beta)^2\cos(\gamma)^2(m_R+2m)(1-s_1) \\
&\quad \frac{1}{4}r^2(1-s_2)\sin(\gamma)^2(m_R+2m)(1-s_1) \\
&\quad -\frac{r^2(1-s_2)\left(I_R+2I_z+2mH^2\right)(1-s_1)}{H^2} \\
b_5 &= r(1-s_2)\left[0,5F_w\sin(\gamma)-W_{t2}-0,5P_u-0,5F_D-0,5G\sin(\gamma)\right] \\
&\quad -\frac{M_p r(1-s_2)}{H}
\end{aligned}$$

where:

r– the radius of the drive wheel replacement,
H – distance between the axes of the tracks,
P_u – pulling force,
m – track mass,
m_R – frame mass,
I_R, I_z, I_x – inertia moment for the robot frame,
β – angle of the robot frame (Fig. 8a),
γ – angle of driveway (Fig. 8a),
W_t – the force of resistance of the rolling track,
F_w – hydrostatic force,
F_D – hydrostatic resistance force,
α_1 – angle of rotation of wheel 1,
α_2 – angle of rotation of wheel 2,
s_1 – slip for wheel 1,
s_2 – slip for wheel 2.

Equation (4) were written down in state space:

$$\dot{\alpha} = A\alpha + B\left[f(\alpha,\beta,\gamma) + G(\alpha,\beta,\gamma)u(t)\right] \tag{5}$$

Because functions $f(\alpha,\beta,\gamma)$ and $G(\alpha,\beta,\gamma)$ don't have the linear form with regard to parameters, there are some inaccuracies in the modelling. The identification system takes the form

$$\dot{\hat{\alpha}} = A\hat{\alpha} + B\left[\hat{f}\left(\alpha,\hat{\beta},\hat{\gamma}\right) + \hat{G}\left(\alpha,\hat{\beta},\hat{\gamma}\right)u\right] + K\tilde{\alpha} \tag{6}$$

where vector $\hat{\alpha}$ is an estimation of the state vector α, $\hat{f}\left(\alpha,\hat{\beta},\hat{\gamma}\right)$, $\hat{G}\left(\alpha,\hat{\beta},\hat{\gamma}\right)$ are estimations of the non-linear functions in equation (5). Accepting the error of the estimation of the state vector in the form

$$\tilde{\alpha} = \alpha - \hat{\alpha} \tag{7}$$

and subtraction equation (6) from equation (5) a description of the identification system in the error space is acquired

$$\dot{\tilde{\alpha}} = A_H\tilde{\alpha} + B\left[\tilde{f}\left(\alpha,\beta,\gamma,\hat{\beta},\hat{\gamma}\right) + \tilde{G}\left(\alpha,\beta,\gamma,\hat{\beta},\hat{\gamma}\right)u\right] \tag{8}$$

where: $A_H = A - K$ and the matrix K is that the characteristic equation of the A_H matrix is strictly stable.

3. Identification using neural networks

Another kind of the solution to the task of identification is an application of artificial neural networks.

Adding and subtracting the expression $A_m\alpha$ from equation (5) where A_m is a stable design matrix [8], we receive

$$\dot{\alpha} = A_m\alpha + (A - A_m)\,\alpha + B\,[f\,(\alpha,\beta,\gamma) + G\,(\alpha,\beta,\gamma)\,u] \tag{9}$$

Equation (8) defines the series–parallel structure of the identification system which is in the form

$$\dot{\hat{\alpha}} = A_m\hat{\alpha} + (A - A_m)\,\alpha + B\left[\hat{f}\left(\alpha,\hat{\beta},\hat{\gamma}\right) + \hat{G}\left(\alpha,\hat{\beta},\hat{\gamma}\right)u\right] \tag{10}$$

where:

$\hat{\alpha}$ is the estimator of the vector of the state α, $\hat{f}\left(\alpha,\hat{\beta},\hat{\gamma}\right)$ and $\hat{G}\left(\alpha,\hat{\beta},\hat{\gamma}\right)$ are estimators of non–linear functions from equation (9).

The error of the estimation of the state is given in the form (7).

Substracting equation (10) subtract from equation (9), a description of the task of identification is received in the error space

$$\dot{\tilde{\alpha}} = A_m\tilde{\alpha} + B\left[\tilde{f}\left(\alpha,\beta,\gamma,\hat{\beta},\hat{\gamma}\right) + \tilde{G}\left(\alpha,\beta,\gamma,\hat{\beta},\hat{\gamma}\right)u\right] \tag{11}$$

where:

$$A_m\tilde{\alpha} = A_m\alpha - A_m\hat{\alpha} \tag{12}$$

$$\tilde{f}\left(\alpha,\beta,\gamma,\hat{\beta},\hat{\gamma}\right) = f\,(\alpha,\beta,\gamma) - \hat{f}\left(\alpha,\hat{\beta},\hat{\gamma}\right) \tag{13}$$

$$\tilde{G}\left(\alpha,\beta,\gamma,\hat{\beta},\hat{\gamma}\right) = G\,(\alpha,\beta,\gamma) - \hat{G}\left(\alpha,\hat{\beta},\hat{\gamma}\right) \tag{14}$$

To determine the function $\hat{f}\left(\alpha,\hat{\beta},\hat{\gamma}\right)$ and $\hat{G}\left(\alpha,\hat{\beta},\hat{\gamma}\right)$ neural networks have been applied.

Since the functions $f\,(\alpha,\beta,\gamma)$ and $G\,(\alpha,\beta,\gamma)$ are supposed to be approximated by neural networks, then

$$f\,(\alpha,\beta,\gamma) = W_f^T S_f\,(\alpha,\beta,\gamma) + \varepsilon_f\,(\alpha,\beta,\gamma) \tag{15}$$

$$G\,(\alpha,\beta,\gamma) = W_G^T S_G\,(\alpha,\beta,\gamma) + \varepsilon_G\,(\alpha,\beta,\gamma) \tag{16}$$

where:

$\varepsilon_f\,(\alpha,\beta,\gamma)$ and $\varepsilon_G\,(\alpha,\beta,\gamma)$ – are the inaccuracies of approximation of the function $f\,(\alpha,\beta,\gamma)$ and $G\,(\alpha,\beta,\gamma)$ through neural networks,

W_f and W_G – matrices of weights of neural connections,

$S_f\,(\alpha,\beta,\gamma)$ and $S_G\,(\alpha,\beta,\gamma)$ – vectors of base functions.

These networks have the structure of the network with the radial functional extension in form of Gauss' function

$$S_j(x) = \exp\left(-\beta \left\| x - c_j \right\|^2\right) \tag{17}$$

where c_j is j–th centre.

A general structure of the system is shown in (Fig. 2):

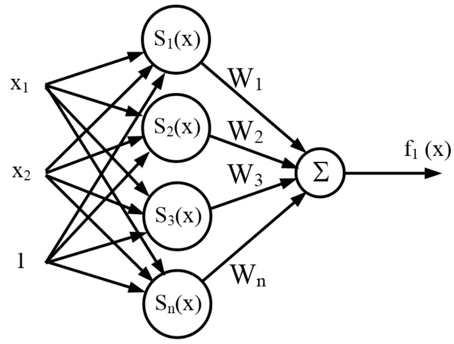

Figure 2 Structure of radial networks approximating functions $\hat{f}\left(\alpha, \hat{\beta}, \hat{\gamma}\right)$ and $\hat{G}\left(\alpha, \hat{\beta}, \hat{\gamma}\right)$

Setting the estimations of functions in equation (13) and (14) in the form

$$\hat{f}\left(\alpha, \hat{\beta}, \hat{\gamma}\right) = \hat{W}_f^T S_f\left(\alpha, \hat{\beta}, \hat{\gamma}\right) \tag{18}$$

$$\hat{G}\left(\alpha, \hat{\beta}, \hat{\gamma}\right) = \hat{W}_G^T S_G\left(\alpha, \hat{\beta}, \hat{\gamma}\right) \tag{19}$$

Formulas (13) and (14) are written in the form

$$\tilde{f}\left(\alpha, \beta, \gamma, \hat{\beta}, \hat{\gamma}\right) = \tilde{W}_f^T S_f\left(\alpha, \beta, \gamma, \hat{\beta}, \hat{\gamma}\right) + \varepsilon_f\left(\alpha, \beta, \gamma\right) \tag{20}$$

$$\tilde{G}\left(\alpha, \beta, \gamma, \hat{\beta}, \hat{\gamma}\right) = \tilde{W}_G^T S_G\left(\alpha, \beta, \gamma, \hat{\beta}, \hat{\gamma}\right) + \varepsilon_G\left(\alpha, \beta, \gamma\right) \tag{21}$$

where:

$\varepsilon_f\left(\alpha, \beta, \gamma\right)$ and $\varepsilon_G\left(\alpha, \beta, \gamma\right)$ – are the errors of approximation of the network,

$\tilde{W}_f$ and $\tilde{W}_G$ – errors of the estimation of weights of the network.

Then equation (11) will be in the form

$$\dot{\tilde{\alpha}} = A_m\tilde{\alpha} + B\left[\tilde{W}_f^T S_f\left(\alpha, \beta, \gamma, \hat{\beta}, \hat{\gamma}\right) + \tilde{W}_G^T S_\Delta\left(\alpha, \beta, \gamma, \hat{\beta}, \hat{\gamma}\right)\right] + B\left[R_f + R_G\right] \tag{22}$$

where:

$$R_f = \varepsilon_f\left(\alpha, \beta, \gamma\right), \quad R_G = \varepsilon_G\left(\alpha, \beta, \gamma\right)u, \quad S_\Delta\left(\alpha, \beta, \gamma, \hat{\beta}, \hat{\gamma}\right) = u \otimes S_G\left(\alpha, \beta, \gamma, \hat{\beta}, \hat{\gamma}\right)$$

The stability of the system was checked according to the Lyapunov stability criterion. It is known that the dynamic system will be stable if a Lyapunov function exists for it [4, 8, 13, 15].

A function was assumed in the form:

$$V = \frac{1}{2}\tilde{\alpha}^T P\tilde{\alpha} + \frac{1}{2}tr\tilde{W}_f^T F_f^{-1}\tilde{W}_f + \frac{1}{2}tr\tilde{W}_G^T F_G^{-1}\tilde{W}_G \tag{23}$$

If this function is to be the Lyapunov function, its derivative has to be negative

$$\begin{aligned} \dot{V} &= -\tilde{\alpha}^T Q\tilde{\alpha} + \tilde{\alpha}^T PB \left[\begin{array}{l} \tilde{W}_f^T S_f\left(\alpha,\beta,\gamma,\hat{\beta},\hat{\gamma}\right) + \\ +\tilde{W}_G^T S_\Delta\left(\alpha,\beta,\gamma,\hat{\beta},\hat{\gamma}\right) + \\ +R_f + R_G \end{array} \right] \\ &+tr\tilde{W}_f^T F_f^{-1}\dot{\tilde{W}}_f + tr\tilde{W}_G^T F_G^{-1}\dot{\tilde{W}}_G \end{aligned} \tag{24}$$

The training of the neural network weights has been carried out according to the formula

$$\dot{\tilde{W}}_f = -F_f S_f\left(\alpha,\beta,\gamma,\hat{\beta},\hat{\gamma}\right)\tilde{\alpha}^T PB \tag{25}$$

$$\dot{\tilde{W}}_G = -F_G S_\Delta\left(\alpha,\beta,\gamma,\hat{\beta},\hat{\gamma}\right)\tilde{\alpha}^T PB \tag{26}$$

From the matrix form of the Lyapunov equation

$$E^T P + PE = -Q = -I \tag{27}$$

a Hertmitian matrix was determined as

$$P = \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} \tag{28}$$

by solving the equation

$$\begin{bmatrix} e_{11} & e_{21} \\ e_{12} & e_{22} \end{bmatrix}\begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} + \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix}\begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \tag{29}$$

A denotation has been assumed

$$h = PB \tag{30}$$

where

$$h = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \tag{31}$$

Finally, the weight training algorithm for (25) and (26) has the form

$$\dot{\hat{W}}_f = F_f S_f\left(\alpha,\beta,\gamma,\hat{\beta},\hat{\gamma}\right)\tilde{\alpha}^T h \tag{32}$$

$$\dot{\hat{W}}_G = F_G S_G\left(\alpha,\beta,\gamma,\hat{\beta},\hat{\gamma}\right)\tilde{\alpha}^T h \tag{33}$$

The identification of the mathematical model of the underwater robot was carried out according to this procedure.

4. The simulation using fuzzy logic

The verification was carried out on a prototype of the underwater robot. We may expect that the estimated model will be different from the mathematical model [8].

To determine functions $\hat{f}\left(\alpha,\hat{\beta},\hat{\gamma}\right)$, $\hat{G}\left(\alpha,\hat{\beta},\hat{\gamma}\right)$ fuzzy logic systems were created in the application Matlab^{TM} (Fig. 3), which makes it possible to create models of fuzzy logic (*fuzzy logic toolbox)* [2, 14].

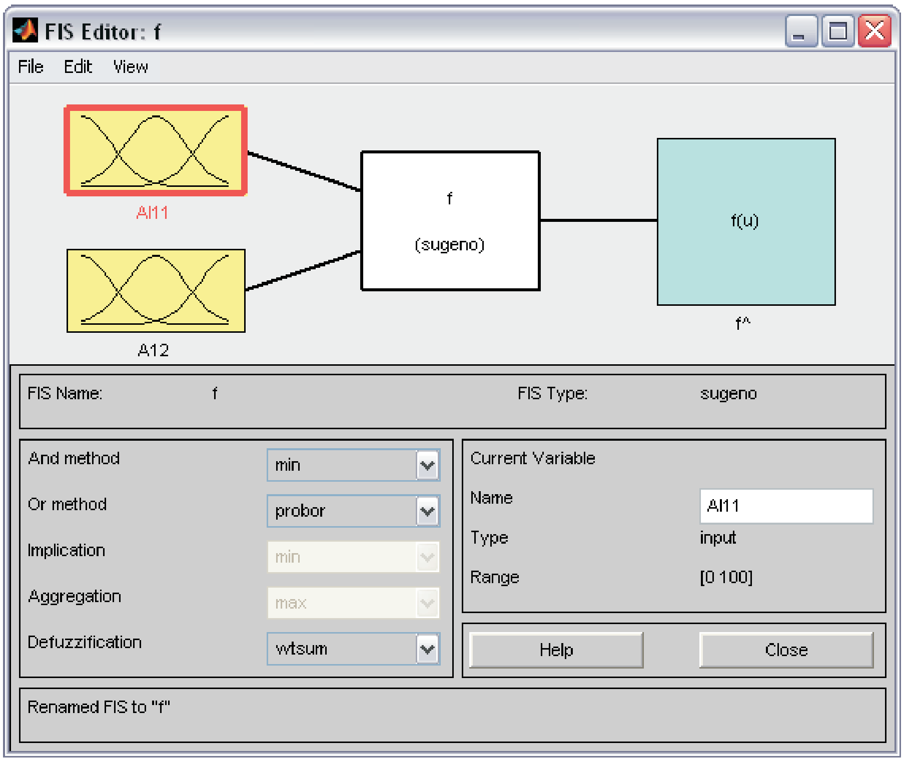

Figure 3 Model of fuzzy logic approximated non–linear functions

The task of the fuzzy logic system is to determine functions $\hat{f}\left(\alpha,\hat{\beta},\hat{\gamma}\right)$, $\hat{G}\left(\alpha,\hat{\beta},\hat{\gamma}\right)$ in such a way, that an error $\tilde{\alpha}$ between the state vector α of the computing model and the estimated state vector $\hat{\alpha}$ is the smallest. Takagi–Sugeno's model was applied in the designing phase [2, 12, 14]. The fuzzification block transformsm the input space in form $X = [\dot{\alpha}_{1a}, \dot{\alpha}_{1b}] \times [\dot{\alpha}_{2a}, \dot{\alpha}_{2b}] \subset R^n$ into fuzzy set $A \in X$ characterised by the membership function $\mu_A(x) : X \to [0,1]$, which assigns a degree of affiliation into fuzzy sets. In (Fig. 4) the membership functions are presented in the form of Gauss' function (gaussmf) according to the input range: $\dot{\alpha}_1 \in [0,100]$, $\dot{\alpha}_2 \in [0,10]$.

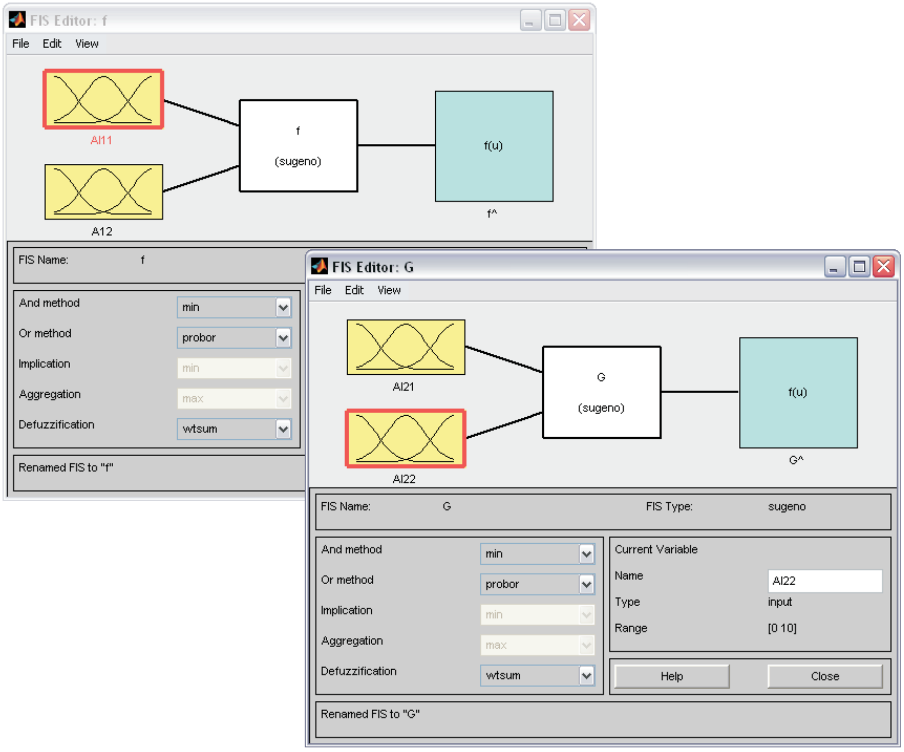

Figure 4 Functions of affiliation and intervals of variability

The base of rules for the model description was accepted as in (Fig. 5). Three membership functions were accepted for the inputs of the fuzzy system and 9 rules of inferring were created. A principle was offered: every rule from one input with every rule of the other input, since the information about each output from the fuzzy systems is missing.

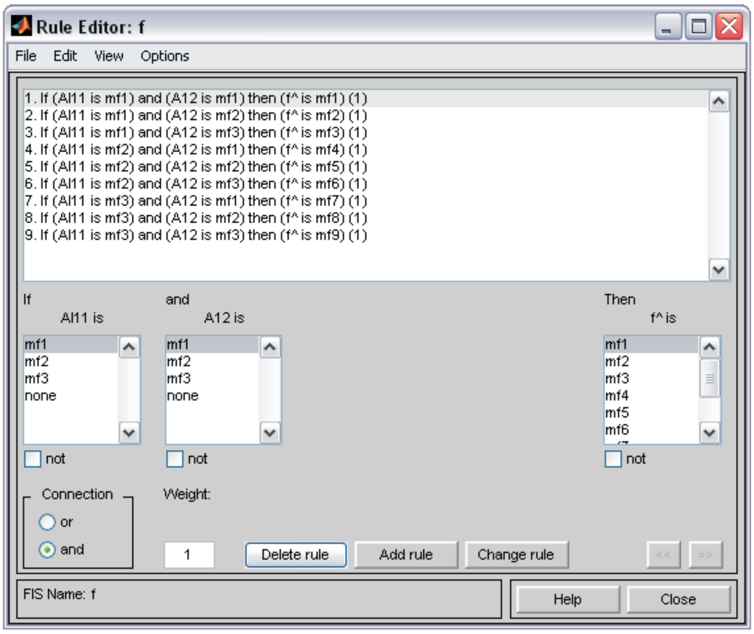

Figure 5 Base of rules for the accepted set

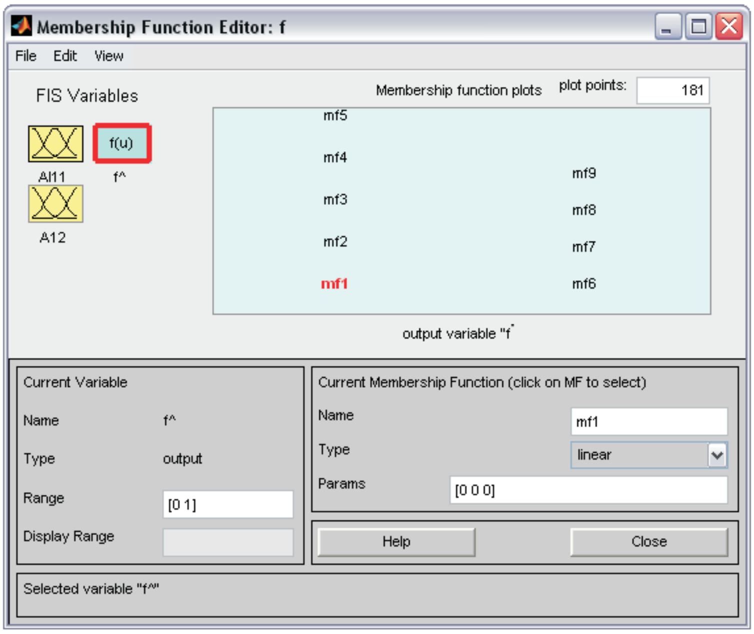

Figure 6 Exit of the fuzzy logic system

The set A was accepted on the input with T–norm [3] of the minimum type

$$\mu_{A_1^j \times \ldots \times A_n^j}(x) = \min\left[\mu_{A_1^j}, \ldots, \mu_{A_n^j}\right] \tag{34}$$

On the output of the Takagi-Sugeno model presented in (Fig. 6) a signal was received

$$y(x) = \frac{\sum\limits_{j=1}^{M} \bar{y}_j \tau_j}{\sum\limits_{j=1}^{M} \tau_j} \tag{35}$$

where:

$$\tau_j = \prod_{i=1}^{n} \mu_{A_i^j}(x_i) \tag{36}$$

is the ignition level of the j–rule.

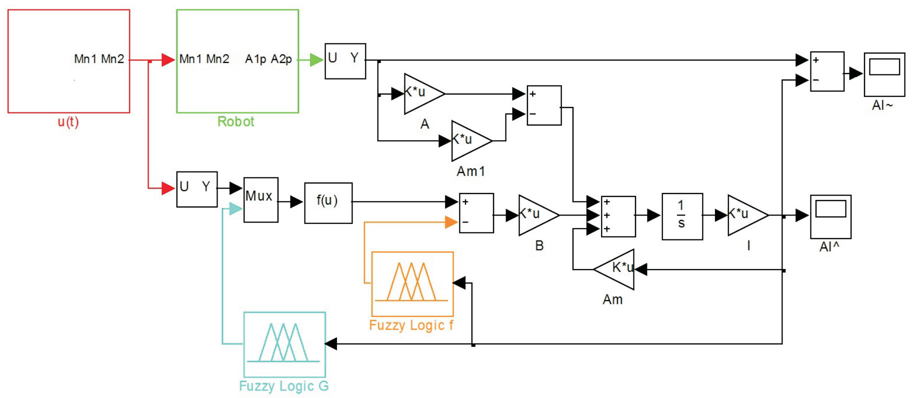

Figure 7 Structure of identification with fuzzy logic

The described fuzzy logic systems were applied for approximation of non–linear functions and they were modeled in the form (Fig. 7).

Data it:

r = 0,02794 m;

H = 0,145 m;

L = 0,127 m;

n = 8;

i = 500/7;

η = 0,45;

P_u = 20 N;

m = 2,8 kg;

m_R = 3 kg;

I_R = 0,008854 kgm^2;

$I_z = 0{,}000651$ kgm^2;
$I_x = 0{,}000059$ kgm^2;
$s_1 = 0$;
$s_2 = 0$.

Fuzzy Logic sets f and G are responsible for approximation of non–linear function. All fuzzy sets use numerical information which explicitly connects the input and output signals.

3D trajectory (Fig. 8a) and velocity of point C robot V_c (Fig. 8b, Fig. 1c)

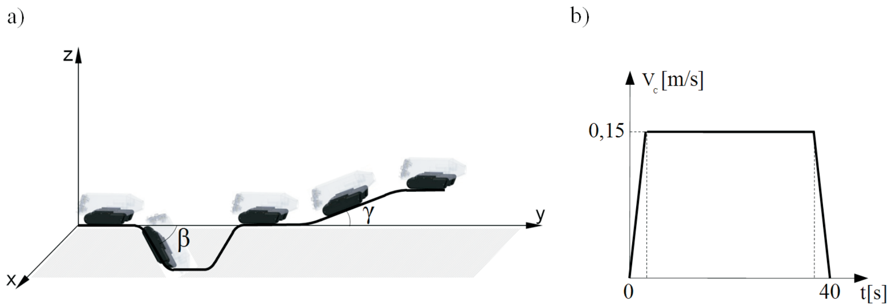

Figure 8 a) 3D trajectory, b) velocity of point C robot

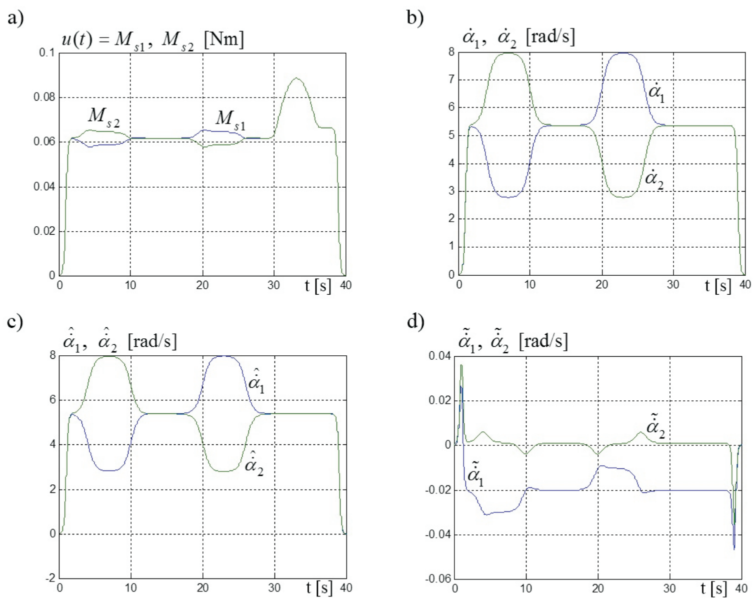

Figure 9 Results of identification: a) input signal, b) angular velocity on the shafts driving motors, c) parameters estimator, d) errors estimator

In the next stage, the parameter identification of the underwater robot was carried out according to the structure (Fig. 7) designed in Matlab^{TM}–Simulink software, taking as the input function $u(t)$ torque of the motors (Fig. 9a).

Moments on the motor shaft received from the inverse dynamics was taken as an input function (Fig. 9a) and than fuzzy identification of underwater robot parameters was carried out according to the chapter 2. Estimated parameters are the angular velocity (Fig. 9c) of the motor shaft, which were compared with the parameters obtained during simulation the inverse kinematics (Fig. 9b). Subtracting them, the angle velocity estimation error were obtained as (7) (Fig. 9d). It can be seen that maximum the estimation error for the angle velocity $\tilde{\dot{\alpha}}$ of the motor shaft is 0,5% (Fig. 9d) compared to $\hat{\dot{\alpha}}$ (Fig. 9c). Obtained solutions of identification fuzzy logic are limited, and the proposed procedure is enabling identification of non–linear systems by applying fuzzy logic systems.

5. The simulation using neural networks

In the next stage identification of the robot parameters was done with the application of neural networks according to the structure (Fig. 10) given for input function $u(t)$ torque of the motors (Fig. 11a).

Data it:
$r = 0{,}02794$ m;
$H = 0{,}145$ m;
$L = 0{,}127$ m;
$n = 8$;
$i = 500/7$; $\eta = 0{,}45$;
$P_u = 20$ N;
$m = 2{,}8$ kg;
$m_R = 3$ kg;
$I_R = 0{,}008854$ kgm^2;
$I_z = 0{,}000651$ kgm^2;
$I_x = 0{,}000059$ kgm^2;
$s_1 = 0$;
$s_2 = 0$.

On the schema (Fig. 10) "Neural network f" and "Neural network G" are models from the chapter 3.

Taking the moments of the motor shaft from the inverse dynamics as an input function (Fig. 11a) the neural identification of the underwater robot parameters was carried out in accordance with the procedure in chapter 3. Estimated parameters are the angular velocity (Fig. 11c) of the motor shaft, were compared to the parameters obtained during simulation the inverse kinematics (Fig. 11b). Subtracting them, the angle velocity estimation error were obtained as (7) (Fig. 11d). It can be seen that maximum the estimation error for the angle velocity $\tilde{\dot{\alpha}}$ of the motor shaft is 0,5% (Fig. 11d) compared to $\hat{\dot{\alpha}}$ (Fig. 9c). Obtained solutions are limited, and the proposed procedure is enabling identification of non-linear systems by applying neural networks.

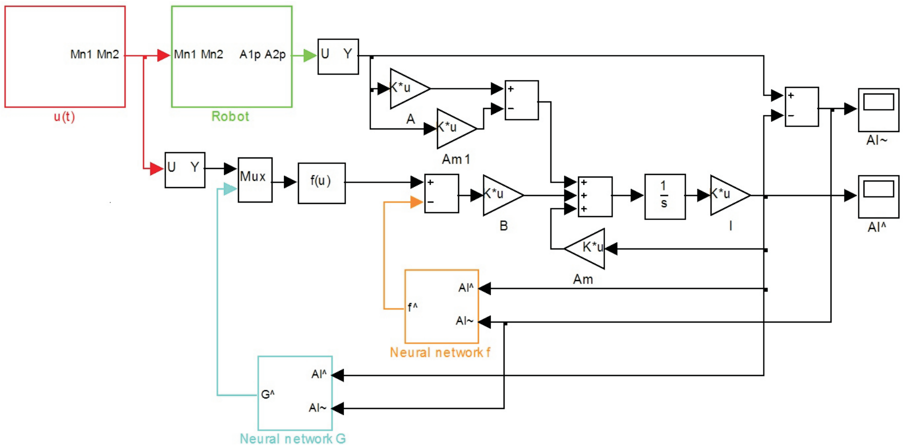

Figure 10 Structure of identification with neural networks

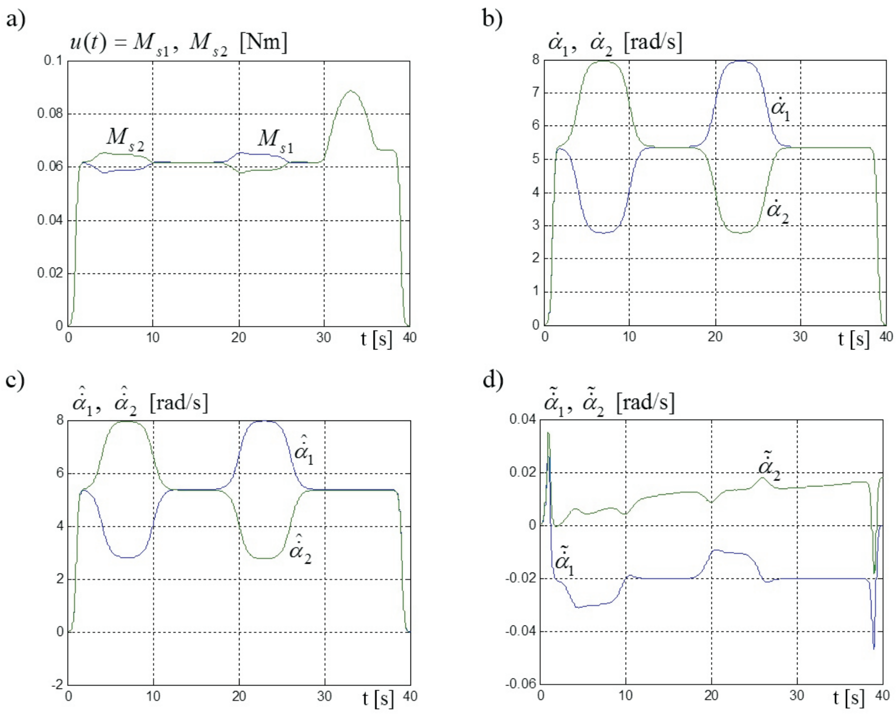

Figure 11 Results of identification. a) input signal, b) angular velocity on the shafts driving motors, c) parameters estimator, d) errors estimator

6. Summary

After carrying out the stages of identification with neural networks and with fuzzy logic it is possible to assume that these methods can be successfully applied during identification of dynamic motion equations, and actual parameters as well as for monitoring dynamic load and detecting damage.

Since results are similar, there a question arise, which method shall we apply during identification of non–linear systems? The answer is simple: system which is more suitable for us or that we implement faster.

The work is a part of the research project N N501 054440

References

[1] **Buratowski, T. and Giergiel, J.**: Dynamics Modeling and Identification of the Amigobot Robot, *Mechanics and Mechanical Engineering*, Vol. 14, No. 1, 65–79, **2010**.

[2] **Buratowski, T., Uhl, T. and Żylski, W.**: The Comparison of parallel and serial–parallel structures of mobile robot Pioneer 2DX state emulator, *Materials of the 7th International IFAC Symposium on Robot Control – SYROCO*, **2003**.

[3] **Driankow, D., Hellendoorn, H. and Reinfrank, M.**: Wprowadzenie do sterowania rozmytego, *WNT*, Warszawa, **1996**.

[4] **Giergiel, J., Hendzel, Z. and Żylski, W.**: Kinematyka, dynamika i sterowanie mobilnych robotów kołowych w ujęciu mechatronicznym, *Monografia, Wydz. IMiR* , AGH Kraków, **2000**.

[5] **Giergiel, J. and Kurc, K.**: Identification of the mathematical model of an inspection mobile robot with fuzzy logic systems and neural networks, *Journal of Theoretical and Applied Mechanics*, 49, 1, 209–225, Warsaw, **2011**.

[6] **Giergiel, M., Buratowski, T., Małka, P., Kurc, K., Kohut, P. and Majkut, K.**: The Project of Tank Inspection Robot, *Key Engineering Materials*, Vol. 518, **2012**.

[7] **Giergiel, M., Buratowski T., Małka, P. and Kurc, K.**: The Mathematical Description of the Robot for the Tank Inspection, *Mechanics and Mechanical Engineering*, Vol. 15, No. 4, 53–62, **2011**.

[8] **Giergiel, J., Kurc, K. and Giergiel, M.**: Mechatroniczne projektowanie robotów inspekcyjnych, *Oficyna Wydawnicza Politechniki Rzeszowskiej*, **2010**.

[9] **Kurc, K. and Szybicki, D.**: Kinematics of a Robot with Crawler Drive, *Mechanics and Mechanical Engineering*, Vol. 15, No. 4, 93–100, **2011**.

[10] **Mężyk, A., świtoński, E., Kciuk, S. and Klein, W.**: Modelling and Investigation of Dynamic Parameters of Tracked Vehicles, *Mechanics and Mechanical Engineering*, Vol. 15, No. 4 115–130, **2011**.

[11] **Osowski, S.**: Sieci neuronowe w ujęciu algorytmicznym, *WNT*, Warszawa, **1996**.

[12] **Rutkowska, D., Piliński, M. and Rutkowski, L.**: Sieci neuronowe, algorytmy genetyczne i systemy rozmyte, *PWN*, Warszawa, **1997**.

[13] **Stefański, A., Kapitaniak, T., Dąbrowski, A.**: The Largest Lyapunov Exponent of Dynamical Systems with Time Delay, *IUTAM Symposium on Chaotic Dynamics and Control of Systems and Processes in Mechanics*, Solid Mechanics and its Applications, V. 122, 493–500, **2005**.

[14] **The MathWorks**, Inc., Fuzzy Logic Toolbox.

[15] **Wojewoda, J., Stefański, A., Wiercigroch, M. and Kapitaniak, T.**: Estimation of Lyapunov exponents for a system with sensitive friction model, *Archive of Applied Mechanics*, V. 79, No. 6–7, 667–677, **2009**.